DOT/FAA/CT- 05/06

Federal Aviation Administration
William J. Hughes Technical Center
Atlantic City International Airport, NJ 08405

Conference Control System Computer-Human Interface Prototype Description and Design Rationale

Shantanu Pai, L-3 Communications, Titan Corporation
Kenneth Allendoerfer, ATO-P

August 2005

Technical Report

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT/FAA/CT-05/06		

4. Title and Subtitle	5. Report Date
Conference Control System Computer-Human Interface Prototype Description and Design Rationale	August 2005
	6. Performing Organization Code ATO-P

7. Author(s)	8. Performing Organization Report No.
Shantanu Pai, L-3 Communications, Titan Corporation Kenneth Allendoerfer, ATO-P	DOT/FAA/CT-05/06

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Federal Aviation Administration NAS Human Factors Group William J. Hughes Technical Center Atlantic City International Airport, NJ 08405	11. Contract or Grant No.

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Federal Aviation Administration Voice Switching and Recording Product Team 800 Independence Ave., S.W. Washington, DC 20591	Technical Report
	14. Sponsoring Agency Code AND-320

15. Supplementary Notes:

16. Abstract

The Federal Aviation Administration (FAA) Air Traffic Control System Command Center (ATCSCC) is responsible for the strategic aspects of the National Airspace System (NAS). The ATCSCC modifies traffic flow and rates when congestion, weather, equipment outages, runway closures, or other operational conditions affect the NAS. Controllers at the ATCSCC accomplish these tasks by communicating with NAS stakeholders like local FAA facilities, airlines, and other national civil aviation authorities.

In 2004, the FAA deployed the Conference Control System (CCS) as part of infrastructure modernization to meet increased capacity demands. The CCS provides many new functions and a computer-human interface (CHI) based on touch-entry display (TED) technology. The NAS Human Factors Group conducted a user-centered design project to explore the CCS CHI requirements. In collaboration with the CCS User Team, we developed mouse- and TED-based CHI prototypes to demonstrate the potential CCS functionality. This report discusses the approach we took in designing the CCS prototype and the rationale for each of the important CHI elements. Many of the concepts developed in the prototype were implemented into the operational CCS. The report also discusses the role of iterative prototyping in increasing designers' and users' understanding of the tasks, requirements, and CHI development process. Future programs can use the design rationale to guide the creation of CHIs for new telecommunication systems. We believe that the design approach adopted in this project allowed for a better elicitation of the user requirements and helped educate the user team regarding human factors and usability issues.

17. Key Words	18. Distribution Statement
Air Traffic Control System Command Center Prototyping Touch-Entry Display	This document is available to the public through the National Technical Information Service, Springfield, Virginia, 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	41	

Table of Contents

List of Illustrations

Figures Page

Tables Page

Executive Summary

The Federal Aviation Administration (FAA) Air Traffic Control System Command Center (ATCSCC) is responsible for the strategic aspects of the National Airspace System (NAS). The ATCSCC modifies traffic flow and rates when congestion, weather, equipment outages, runway closures, or other operational conditions affect the NAS. Controllers at the ATCSCC accomplish these tasks by communicating with NAS stakeholders like local FAA facilities, airlines, and other national civil aviation authorities.

The Operational Telephone System (OTS) was the legacy telecommunications system that provided the conferencing functions. In 2004, the FAA replaced the OTS with the Conference Control System (CCS) as part of infrastructure modernization to meet increased capacity demands. The CCS provides many new functions and a computer-human interface (CHI) based on touch-entry display (TED) technology.

In 2002, the Product Team for Voice Switching Recording (AND-320) tasked the NAS Human Factors Group to conduct a user-centered design project to explore the CCS CHI requirements. In collaboration with the CCS User Team, we developed a mouse- and a TED-based CHI prototype to demonstrate the potential CCS functionality.

This report discusses the approach we took in designing the CCS prototype and the rationale for each of the important CHI elements. Many of the concepts developed in the prototype were implemented into the operational CCS. The report also discusses the role of iterative prototyping in increasing designers' and users' understanding of the tasks, requirements, and CHI development process. Future programs can use the design rationale to guide the creation of CHIs for new telecommunication systems.

We believe that the design approach adopted in this project allowed for a better elicitation of the user requirements and helped educate the user team regarding human factors and usability issues.

1. INTRODUCTION

The Federal Aviation Administration (FAA) Air Traffic Control System Command Center (ATCSCC) is responsible for the strategic aspects of the National Airspace System (NAS). Controllers at the ATCSCC lead and coordinate the FAA effort to modify traffic routes and flows when congestion, weather, equipment outages, runway closures, or other operational conditions that affect the NAS. Controllers accomplish these tasks by communicating directly with Air Route Traffic Control Centers (ARTCCs), Terminal Radar Approach Control (TRACON) facilities, towers, airlines, weather units, the military, and other affected organizations such as the civil aviation authorities in Canada and Mexico.

Many of these communications are multiparty teleconferences that ATCSCC controllers initiate and manage. Until 2004, the Operational Telephone System (OTS) console provided conferencing functions at the ATCSCC. The OTS controlled a large bridge that connected conference participants. Controllers used the system to configure, initiate, manage, and terminate conferences. The OTS system used a text-based, menu-driven computer-human interface (CHI). Controllers used cursor keys to navigate among the menus and select parties for actions such as call, mute, hold, and transfer.

The OTS reached the end of its service life and was replaced by the Conference Control System (CCS) in 2004. The CCS provides new functions and a more modern CHI employing a touch-entry display (TED) and elements such as push buttons, pull down menus, and icons.

This document describes the process and outcomes of a CHI development project conducted by the NAS Human Factors Group in 2002 through 2004 on behalf of the CCS Program Office. The project followed user-centered design best practices to explore CHI concepts for the CCS. The project addressed the following three basic questions: What sorts of CHI designs would be most usable by ATCSCC personnel? Should the CCS CHI be similar to the OTS or could other interaction concepts be used? What new functions and features would improve operational performance by making controllers' jobs easier and more efficient? In collaboration with the CCS User Team, we developed mouse- and TED-based versions of a prototype CHI to demonstrate the potential CCS functionality. We did this by meeting with users and conducting iterative design reviews and evaluations. The prototypes allowed user representatives to visualize and provide feedback about the evolving CHI requirements.

This document describes the ATCSCC communication tasks, the CHI prototype we developed, and the rationales for the design decisions that our team made. The prototype described in the document does not correspond directly to the operational CCS as it was ultimately deployed.[1] The purpose of this document is to describe the prototype and the process so that future programs

[1] The operational CCS is a commercial product created by Frequentis USA with modifications to suit the ATCSCC environment. The operational CCS contains many of the concepts explored in the CHI prototype but provides them in somewhat different formats and with many different design details. These differences are indicative of the complexities of developing and acquiring products in a commercial environment. Sometimes the CHI design envisioned during the prototyping turned out to be unavailable in the marketplace or not feasible given budgetary, schedule, or technical constraints. As of January 2005, the CCS has been operational at the ATCSCC for several months and has received generally positive reviews from controllers.

developing telecommunication systems will have the benefit of knowing which CHI options have already been examined and the reasons why some options are preferable from a human factors or operational standpoint.

2. ATCSCC TELECOMMUNICATION FUNCTIONS

The key task of controllers at the ATCSCC is to communicate with ARTCCs, TRACONs, towers, airlines, weather units, and other stakeholders to coordinate responses to situations in the NAS. Nearly all controllers are assigned to work in the East Area, West Area, or the Severe Weather Unit. The East and West Areas are responsible for communication with stakeholders located in the eastern or western halves of the United States. Severe Weather handles national responses to weather situations. The controller workstations and the accompanying communication consoles on the work floor are grouped accordingly.

Depending on the situation, there are several ways in which controllers communicate with stakeholders. Stakeholders can call in seeking information from the ATCSCC or controllers can call out to disseminate information. In the simplest terms, the controllers use their communication system to make and answer telephone calls. If the call is between more than two parties, it is known as a conference.[2] Sometimes a conference is needed because the controller must communicate the same information to many parties. Other times, a conference is needed because other parties must communicate with each other to plan strategies and the controller serves as an intermediary and facilitator.

In the following sections, we describe the most important functions of any communication system for the ATCSCC. Both the OTS and the new CCS contain these functions, though the details between the systems differ. The names and characteristics of the functions are influenced by the controllers' history with communication systems and their operational procedures and practices.

2.1 Telephone Functions

Controllers at the ATCSCC spend much of their time making or answering telephone calls. When making an outgoing call, a controller can manually dial the number using a traditional dial pad (0 through 9, * and #), known as indirect access. However, they normally rely on speed dial buttons for numbers that they call regularly, known as direct access. Because the ATCSCC communicates with hundreds of individual phone numbers, the speed dial buttons must be organized in a phonebook that allows controllers to quickly find parties. For each console, a homepage is established that contains the most frequently used speed dial buttons for that position.

Any teleconferencing system at the ATCSCC needs to provide basic telephone functionality. Most of these functions are not normally available on standard desk phones but are standard functions of any advanced communication system.

- *Hold*. When a call is put on hold, the outside party remains connected to the system but can no longer hear or be heard by the controller. Controllers typically place parties on hold while they confer with other controllers or while they contact other parties who are

[2] From a technical perspective, there is no significant distinction between a two-party call and a three-party conference. Every call is a conference with a variable number of participants.

needed for a conference. Any hold function must indicate to the controller that the party is on hold and allow the controller to take the party off hold.

- *Transfer.* When a call is transferred, it is sent from one controller position to another. Transfer is typically used when an incoming caller has reached the wrong controller or when an outside party has other business with a second controller position once the original discussion is over. Any transfer function must allow the controller to indicate which position the call is to be transferred to.
- *Barge.* This function allows controllers to add themselves to an ongoing call or conference at another controller's position. Controllers normally coordinate barges with the conference host beforehand. Any barge function must allow the controller to specify which console is going to be barged.
- *Override.* This function allows controllers to add themselves to an ongoing call or conference at another controller's position. Unlike barge, only the overridden controller can hear the overriding controller. Override is often used when the overriding controller has information for the overridden controller that outside parties do not need to hear or when the overriding controller does not want to interrupt a discussion on a conference. Any override function must allow the controller to specify which console is to be overridden.
- *Forward.* Sometimes confused with transfer, the forward function sends all incoming calls intended for one position to a different position. This is typically used when a controller is away from the position or busy with other tasks. Any forward function must allow the controller to indicate which console is to receive the forward and to allow the originating console to "unforward" its calls when necessary.

2.2 Progressive Conference

A progressive conference is one that is not planned or scheduled but evolves over the course of a discussion. A progressive conference typically begins as a two-party call. As the discussion proceeds, it becomes clear that other parties need to participate. The controller adds these new parties by placing the first party on hold, contacting the other parties on another line, and then patching all the parties together. Depending on the situation, a progressive conference can evolve into a large conference. If a progressive conference becomes too large, a controller may end the progressive conference and schedule a preset or meet-me conference that contains all the necessary participants.

2.3 Preset Conference

A preset conference (also known as a "blast out" conference) is a planned conference with an established set of participants each of whom has an associated telephone number. It is useful to think of a preset conference as multiparty speed dial: one or two entries by the controller can dial tens or hundreds of phone numbers. The controller hosting the conference initiates a preset conference and the conference bridge dials the number for each participant. The controller can add or remove parties to the conference as necessary but cannot edit the saved list of participants.

Examples of preset conferences at the ATCSCC are the so-called first tier conferences. The Washington ARTCC (ZDC) first tier conference contains ZDC and all the ARTCCs adjacent to it (New York, Boston, Cleveland, Indianapolis, Atlanta, and Jacksonville). When a situation

arises that affects ZDC, the controller may initiate the ZDC first tier preset conference to automatically dial all the affected ARTCCs. If the situation has broader implications, the controller may initiate the ZDC second tier conference which includes ZDC, all the ZDC first tier members, and all the ARTCCs adjacent to the ZDC first tiers.

Preset conferences are established for teleconferences that occur regularly with a fixed membership. These might be conferences that occur each day at a fixed time or for situations that occur frequently. Because a preset conference dials out to a list of numbers, participants are expected to be available at the assigned phone number at the time the conference is scheduled.

2.4 Meet-Me Conference

A meet-me conference is similar to a preset conference in that the participant list is defined ahead of time. The controller schedules and activates the conference, and the participants call in using an access code. Controllers call the access code a personal identification number (PIN), though technically speaking it is not truly personal; we will use their term. Callers cannot join a conference unless they enter the correct PIN and are members of the participant list (or the controller adds them manually). Typically the time, call-in number, and PIN for a meet-me conference are stable and distributed to participants well ahead of time.

2.5 Ad Hoc Conference

An ad hoc conference is similar to a meet-me conference except that the participant list is not defined. The controller establishes an ad hoc conference and assigns it a PIN. The information about the conference is distributed ahead of time and participants join the conference by calling the number and entering the PIN. An ad hoc conference would be used when the situation is unusual or evolving such that no established preset or meet-me conference is suitable. This may be important, especially when conference participants and their accompanying phone numbers change regularly.

3. CHI PROTOTYPING METHOD

In this section, we describe how we created the CCS CHI prototype. We used common user-centered design techniques of iterative rapid prototyping and usability assessments. We developed the prototype to provide the required functionality as described above and to suit the unique aspects of the ATCSCC environment, tasks, and users.

3.1 Participants

Personnel from the ATCSCC participated in the prototype development at several levels of involvement. First, the two facility representatives to the program (one management, one union) served as our primary contacts and subject matter experts. They participated in all development activities and helped coordinate the participation of others. They worked closely with us to create the details of each page and function and helped create test scenarios. Second, three members of the CCS User Team served as regular consultants and participants in usability tests. The CCS User Team was a separate, formal working group that worked directly with the program office on the program requirements and the operational testing of the operational CCS. These five ATCSCC personnel, several technical experts and consultants, plus ourselves serving as designers and developers, constituted our CCS CHI prototype development team.

Finally, many controllers, maintenance technicians, and managers from the ATCSCC participated in one or more of the review sessions, usability tests, and demonstrations that occurred throughout the process.

3.2 Equipment

We created TED-based and mouse-based versions of the CCS prototype interface using Microsoft Visual Basic® on a Pentium II 333 Mhz system with 64 MB of RAM. We developed both versions of the prototype for us on a 15-inch diagonal display. The prototype can operate on displays of other sizes, but the buttons and characters will not appear in the intended size. The TED-based version will operate on regular (i.e., no touch-entry) displays using the mouse to press buttons instead of the TED. The TED was a NEC Multisync LCD 1510.

Copies of the prototype can be obtained from the authors. A script and installation instructions that demonstrate its functionality can be found in Appendix A and Appendix B.

3.3 Iterative Design Process

We developed the CCS CHI prototype using an iterative design process. The prototype underwent six iterations. At the end of the sixth iteration, the team arrived at a consensus on the various CHI elements and functions of the prototype. Figure 1 shows the overall prototype development process.

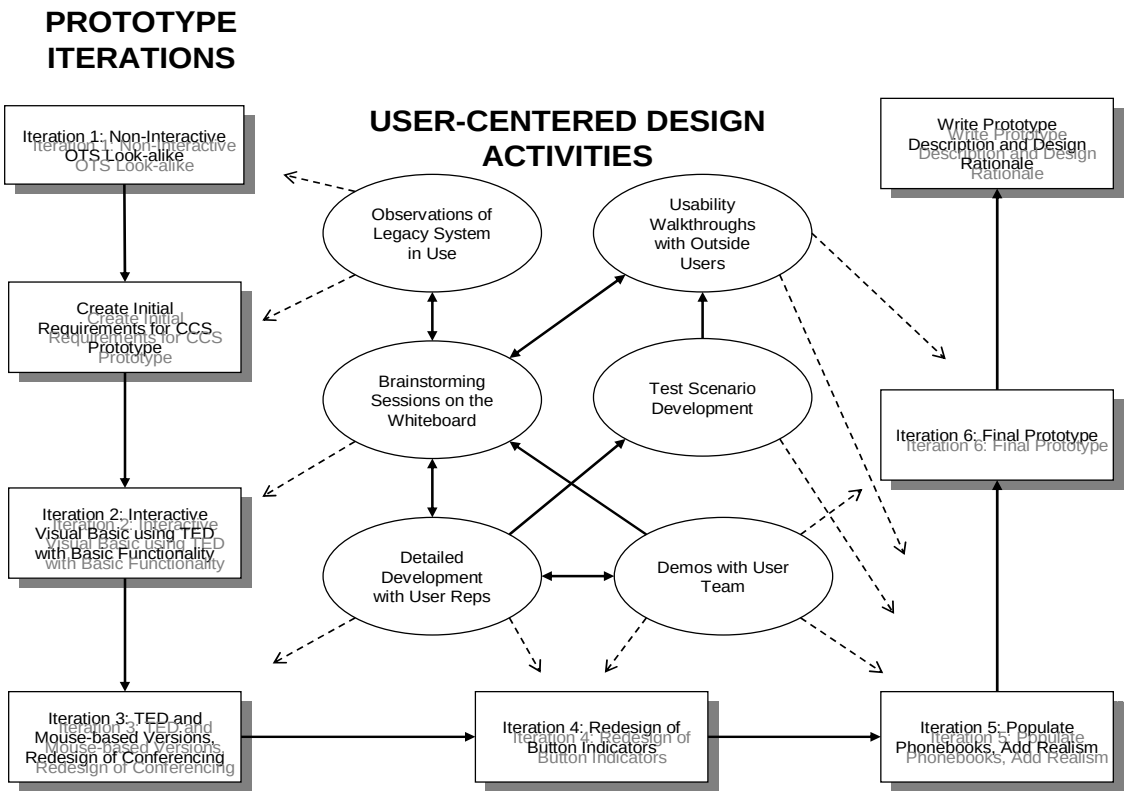


Figure 1. User-centered design activities and prototype iterations.

Prototype iterations are depicted as rectangles. Iterations progress from one to the next with increasing interactivity and realism. Activities are depicted in ovals. There is some overall order to the activities (e.g., a usability walkthrough cannot occur before some ideas have been brainstormed), but the user-centered activities did not (and typically do not) follow a strict linear path. Depending on the outcome of one activity, the team may need to retrace its steps, question earlier decisions, and revise its approach. The dark solid arrows show movement between iterations or activities. The dotted arrows show how the activities provided information for the iterations. For example, Iteration 4 resulted from information learned during detailed development sessions with user representatives and demonstrations with the larger user team.

The OTS formed an initial set of requirements that we built the CCS prototype to meet. We observed controllers using the OTS operationally and discussed its functions at length with the team. We did this to educate ourselves about the controllers' tasks and requirements and to understand what characteristics the CCS would need to have for users to consider it as an improvement over the OTS. In particular, we discussed the characteristics of the OTS that users liked and those that they did not. For example, the controllers believed that the concept of a homepage, where the common speed dial buttons are located, was an essential element of the OTS CHI.

In Iteration 1, we developed a low fidelity prototype using Microsoft PowerPoint that essentially re-created the text-based OTS CHI. Although this was not an interactive prototype, it allowed the team to visualize the initial issues and inspired them to think about the possibilities of a modern CHI. Iteration 1 yielded requirements that any CCS implementation would need to meet, similar to those discussed in Section 2.

We developed Iterations 2 through 6 of the prototype using Microsoft Visual Basic. These versions were interactive and allowed users to exercise the functionality. Because the prototype did not actually control a telecommunications bridge, we built scripted events, such as incoming calls, to allow users to see all aspects of the CHI.

At the beginning of each iteration, we demonstrated the prototype to the user representatives and the CCS User Team. They completed short operational scripts (such as the one provided in Appendix A) to exercise the available functions and highlight changes from the previous iteration. The team members interacted with the prototype using the TED or mouse and provided us with feedback and comments. We consolidated the comments into a set of changes for the next iteration and discussed the changes with the team.

As the iterations progressed, we received more involvement and input from user representatives who were not members of the development team. These users completed the same scripts and provided feedback but with the benefit of a fresh set of eyes. Their feedback helped us understand the occasional differences between the user representatives and the overall user community and served as a validation of the team's decisions.

We conducted five of the six iterations at the ATCSCC in Herndon, VA. Iteration 5 occurred at the Research Development and Human Factors Laboratory at the FAA William J. Hughes Technical Center in Atlantic City International Airport, New Jersey. We dedicated this entire iteration to populating the telephone directories, two-party call buttons, and conference buttons with actual names of real outside parties and phone numbers to make the prototype realistic. This was important to the validity of the usability walkthroughs with users not on the design team. We often found that our usability test participants initially focused on these minor

inconsistencies and oversights rather than broader CHI concepts and usability. By making sure these details were correct, our evaluators could focus on design issues to provide more helpful feedback.

4. CCS PROTOTYPE DESCRIPTION

In the following sections, we describe how the TED-based and mouse-based CCS CHI prototypes appear and operate. In many cases, we also explain why the selected design is advantageous from a human factors or operational perspective.

4.1 Buttons

Buttons are areas on a page that can be selected using the TED or a mouse. They enable users to log on, navigate between and within sections or pages, answer incoming calls, and perform telephone and utility tasks. Buttons have a three-dimensional appearance and seem to be "pressed" when touched or clicked. For demonstration purposes, we programmed the buttons in the TED-based prototype to provide an audible click when touched.

4.1.1 Description

Each button is labeled. In addition, buttons are color coded to provide a redundant cue to help users identify different button types and functionality. The flowing list defines the significance of each color:

- Blue - outside party that is not connected,
- White - parties that are currently connected,
- Yellow - parties on hold or ringing,
- Dark green - incoming calls meant for a different console,
- Light green - navigation buttons, orange indicates telephone functions (e.g., transfer, hold, or forward), and
- Pink - utility buttons.

Color coding is used redundantly with the button labels. Table 1 shows the different states of buttons representing outside parties in which a change in state changes both the color and the label. Table 2 shows other types of buttons in the system that do not change state.

Table 1. Status of Buttons Representing Outside Parties

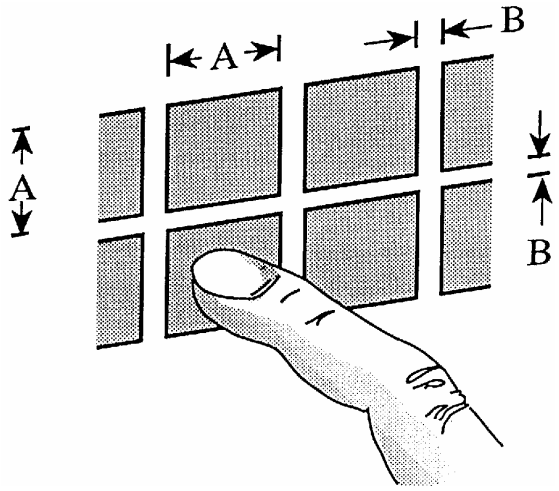
Button	Button State	Description
	Unconnected party	Blue color button with just the party name indicates an outside party that is not connected.
	Connected party	White button color with black text label indicates a party currently connected to the specified console number.
	Party on hold	Yellow button color with black text label indicates the button on hold or ringing. The label indicates console number where the party is being held.
  	Button in an intermediate stage (selected, calling, or ringing)	Yellow color button represents a party that is not connected but selected for further action. For example, to place an outgoing call, users first select the blue button (described above). When selected the button changes its color to yellow. Labels for an outgoing call first change to “Calling XXX” and then to “XXX ringing.” Once connected, the color changes to white and the label indicates that the button is connected. Note that the yellow color is used to indicate an in-between state for a button. For example, a yellow button could mean either a party on hold or a call under process based on the label text.
	Incoming call	Dark green color button represents incoming calls that are meant for a different console. The label indicates to which console to direct the call.

Table 2. Other Button Types in the CCS Prototype

Button	Button Type	Description
	Navigation	Light green indicates a navigation button
	Telephone function	Orange color represents buttons for telephone function such as transfer, hold, or forward.
	Utility function	Pink color represents buttons for utility functions such as volume control, font size, or display control.

4.1.2 Specifications

The size and separation of the buttons for the TED-based prototype follow the FAA Human Factors Design Standard (HFDS) standards for touch input (Ahlstrom & Longo, 2003). Figure 2 is reproduced from Section 9.4.2.4 of that document. Buttons in the TED-based version of the prototype are 0.75 in. × 0.75 in. in size in accordance with the standard.



	A Size	B Separation
Minimum	19 mm (0.75 in.)	3 mm (0.13 in.)
Maximum	38 mm (1.5 in.)	6 mm (0.25 in.)

Figure 2. Button specifications (Ahlstrom & Longo, 2003).

We used a size 8 font for the button labels in both TED-based and mouse-based versions of the prototype. One of the features available in the prototype allows users to adjust the font size. However, the size of the button restricts the maximum size of the font because the button size of 0.75 in. x 0.75 in. remains constant irrespective of the font size. Depending on the number of letters in a particular label, a label with a bigger font size might not fit a 0.75 in. x 0.75 in. button rendering the label illegible (see Figure 3). Hence, users need to choose an appropriate font size to ensure a correct fit. Table 3 shows the font size and corresponding number of characters that can fit on a 0.75 in. x 0.75 in. size button.

To fully accommodate all the labels on a 0.75 in. x 0.75 in. size button, we recommend a size 8 font. We could accommodate a bigger font size by increasing the size of the button. However, given the amount of space available, this would mean fewer buttons on each page.

Font Size: 8 Font Size: 12

Figure 3. Button labels with different sized fonts.

Table 3. Font Size and Number of Characters for Button Labels

Font Size	**Number of upper case letters**	**Number of lower case letters**
8	6	10
10	5	8
12	4	7

4.2 Page Elements

The TED-based version of the interface contains four elements that are accessible on every page: telephone functions, phonebook tabs, incoming call list, and utility functions (see Figure 4). The center portion of the TED is reserved for displaying pages. The homepage is the default page displayed in the center when the user first logs on to the system.

The four basic elements are the buttons for accessing the telephone functions, phonebook tabs, buttons for accessing the preference functions, and the incoming call list area. We describe these in the following sections.

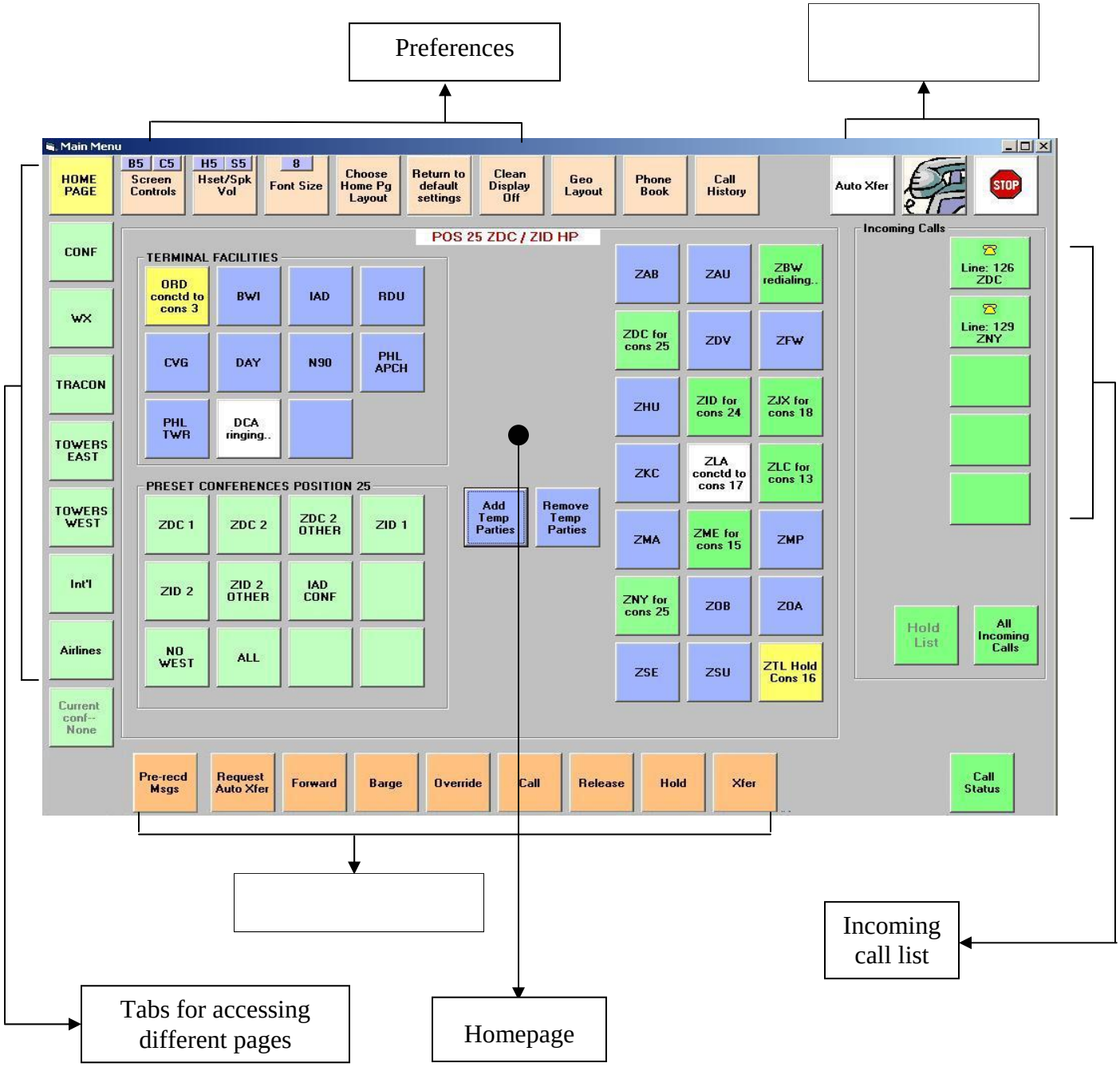


Figure 4. TED-based prototype CHI.

The mouse-based version of the prototype is similar to the TED-based version. We used standard Windows graphical user interface elements such as a tabbed control for accessing the different pages, scroll bars, and so on (see Figure 5). The function of each of the buttons in the mouse-based version is the same as those in the TED-based version.

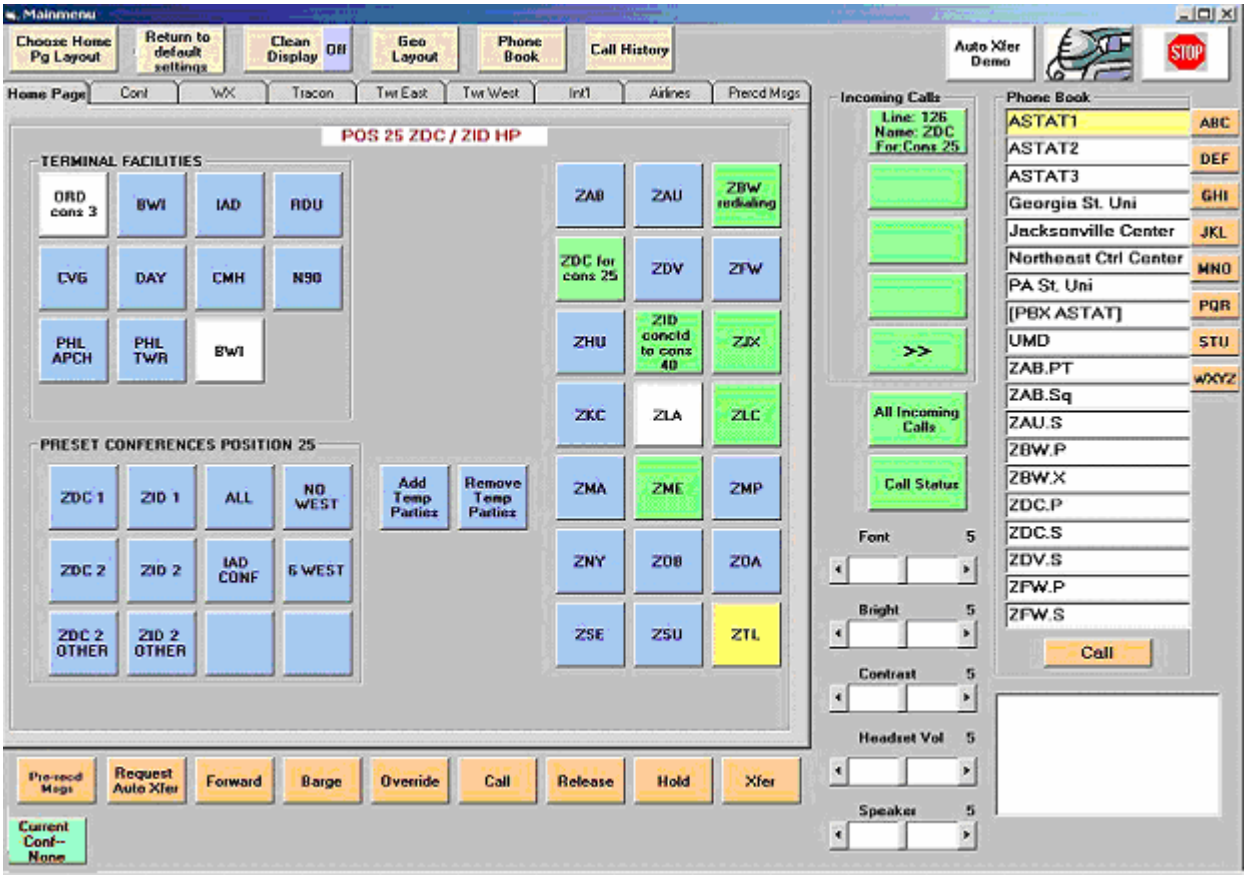


Figure 5. Mouse-based prototype CHI.

4.2.1 Homepage

Each operational position has its own homepage that provides users with access to the most frequently used parties and tasks at that position. For example, a controller working in the Severe Weather Area may need quick access to the weather units at ARTCCs whereas a controller staffing the West Area may need quick access to the Traffic Management Units (TMUs) at the same ARTCCs. For prototyping purposes, we selected an operational position (Console 25) that encompassed the important CCS functionality. The actual CCS contains individual pages customized for each position. During the installation process, the ATCSCC and the vendor developed homepage configurations for each operational position.

A party or conference on the homepage is redundant with other instances of that party or conference elsewhere in the system. For example, the Baltimore-Washington International (BWI) button may appear on the homepage, but it may also appear on the TRACONs page in the phonebook and, if BWI is calling, in the Incoming Call list. This way, frequently contacted parties can be accessed quickly regardless of which page the controller is viewing. It also

ensures that consistent information is being provided about that party everywhere. For example, if the controller is looking at the TRACONs page in the phonebook when BWI calls, the controller should not see a static BWI button on the TRACON page and a flashing BWI button in the Incoming Call list. Redundancy reduces the chances for confusion and provides a consistent picture on every page.

In addition, an operational situation might require frequent communication with a party that is not listed on the current homepage. To access that party, the controller would need to use the phonebook or the dial pad, which would increase workload and frustration. The Add Temp Parties button allows the user to add temporary parties to a homepage. The user first selects the Add Temp Parties button and then selects the desired party from the phonebook. A button appears on the homepage with the selected party as its label. To remove the temporary party, the user selects the temporary party button followed by selecting the Remove Temp Parties button.

4.2.2 Telephone Functions

The nine horizontally arranged buttons at the bottom of the page allow the users to access the telephone functions. Table 4 describes the functionality of each of these buttons in rough order of importance for the TED-based interface.

Table 4. Telephone Function Buttons for the TED-based Interface

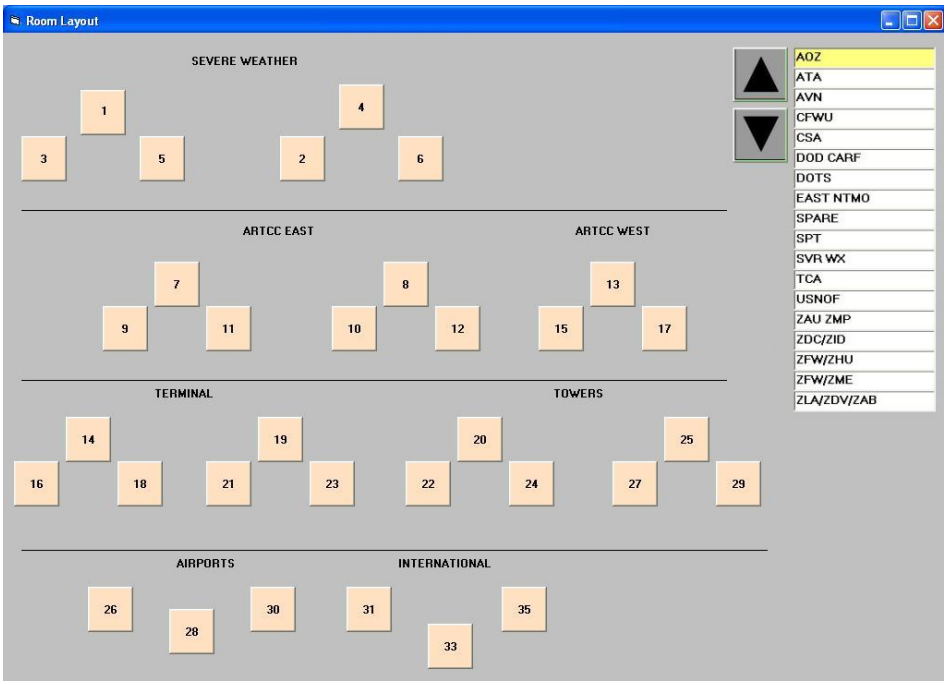
Function Buttons	Description
Xfer	Incoming calls from the Incoming Call list and ongoing two-party calls can be transferred to other consoles using the Xfer button. To transfer a party, the user first selects the party they want to transfer and then selects Xfer. This action displays the console layout page (see Figure 6). The user can either choose the console (represented by a button) or select the party from the in-house list. The button layout follows the same layout as the consoles on the floor. With this option, the user may select the physical location of the intended console to which the call needs to be transferred. The console layout feature may prove useful in instances where ATCSCC controllers know the physical location of the console to which they intend to transfer an incoming call but do not recall its console number. Also, scrolling through the in-house list could be tedious given the number of consoles. The buttons provide layouts of the consoles as well as the numbers associated with them. A drawback associated with the console layout is that the layout in the CCS needs to be updated every time a new console or a position is created on the floor at the ATCSCC. Another way of transferring a call is by directly dialing the console number using the dial pad available in the phonebook. The console layout in the mouse-based interface can also be used as a console status indicator. For example, the status of Console 4 connected to an external party, ZNY, can be indicated by its label "Cons 4 conctd to ZNY." The user group proposed this feature during one of the design iterations. The room layout displayed in Figure 6 does not reflect the actual names and layout of the consoles at the ATCSCC and is a sample for illustration purposes only.  Figure 6. Transfer page.

Table 4. Telephone Function Buttons for the TED-based Interface (continued)

Function Buttons	Description
	Incoming calls or ongoing two party calls can be temporarily put on hold using the Hold button. To do this, the user first selects the party and then selects the Hold button. The button label for the selected party now says "Hold." Selecting the party again releases it from hold.
	The Release button allows users to end ongoing calls. To end an ongoing call, the user selects the intended party followed by the Release button.
	The Call button allows users to place two-party calls by selecting a party followed by selecting the Call button. A progressive conference can also be initiated in a similar fashion by selecting multiple parties before pressing the Call button.
	The Override button allows users to participate in a two-party call or a conference being hosted at a different console. All parties can listen and talk as before with one exception: outside participants cannot hear the overriding console. To override a console, the user selects the Override button to open a keypad followed by choosing the position number to override.
	The Barge button allows users to join an existing two-party call or an ongoing conference being hosted at a different console. To barge, the user first selects an ongoing call and then selects Barge to join as a participant.
	The Forward button allows all incoming calls to be forwarded to a specified position. To forward incoming calls, the user selects the Forward button to open a keypad followed by choosing the position number to forward.
	The Request Auto Xfer button allows users to initiate an auto transfer request on a call in progress at a different console. To do so, the user first selects the ongoing call followed by selecting the Request Auto Xfer button. The console at which the call is in progress receives a message that indicates that another console is in a queue for the outside party.
	This Pre-recd msgs button allows users to create a list of parties to which contingency information may be disseminated simultaneously. To create the list of parties and record a message, the user selects the Pre-recd msgs button to open the contingency list page where the user creates a list of parties from the phonebook and records a message to be sent to these parties. The user then selects the "Call All" button on the contingency list page to send the prerecorded messages to the selected parties simultaneously.

4.2.3 Phonebook Tabs

The first eight vertically arranged buttons on the left side of the page (see Figure 4) can be used to access categorized sections of the phonebook. Some sections include subsections. For example, the conference directory is categorized into preset conferences, ad hoc conferences, meet-me conferences, and progressive conferences. A user can step through different subdirectories to access a particular party or conference. Directory tabs for weather, TRACON, towers, international, and airline provide instant access to the subdirectories or parties listed under its category. Except for the airlines tab, each tab provides buttons representing direct lines to individual parties. The airlines tab lists subdirectories for different types of airlines (e.g., carriers, commuters, cargo, and international). Each airline may have several participants that can be handled collectively as a preset conference.

The Current Conference status button located below the phonebook tabs indicates the type of ongoing conference, if any. Selecting this button provides the user with instant access to the corresponding Conference Management page. We designed this feature to provide the users with one-touch access to current ongoing conferences and to ensure that the users are aware of ongoing conferences. Conferences are discussed in detail in Section 4.3.

4.2.4 Incoming Call List

The incoming call area displays incoming calls to a particular console (see Figure 7). Users can always access incoming calls by selecting the All Incoming Calls button located at the bottom of the incoming call area. Controllers answer incoming calls by selecting the corresponding button. Because of high number of incoming calls and the limited number of buttons that can be accommodated in the incoming call area, the button disappears from the incoming call area once selected. The next incoming call in the area now moves up by one position. However, all ongoing calls, including calls on hold, are displayed in the call status list.

To view the status of all ongoing calls in the system, a user selects the Call Status button located at the bottom right corner of the page. This action opens the Call Status list (see Figure 8) showing the status of all current calls. After answering an incoming call, the user has the choice of transferring it to a different position, putting it on hold, releasing it, or starting a progressive conference by calling additional parties from the directory.

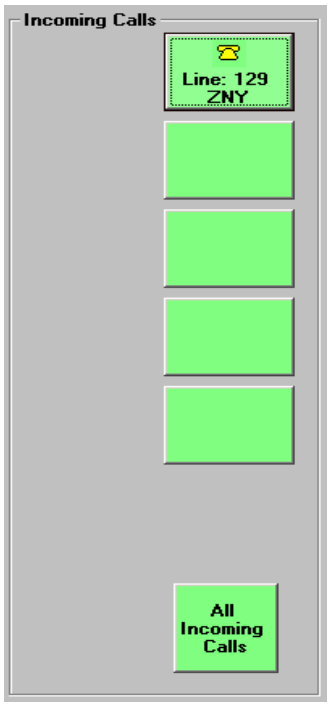


Figure 7. Incoming Call list.

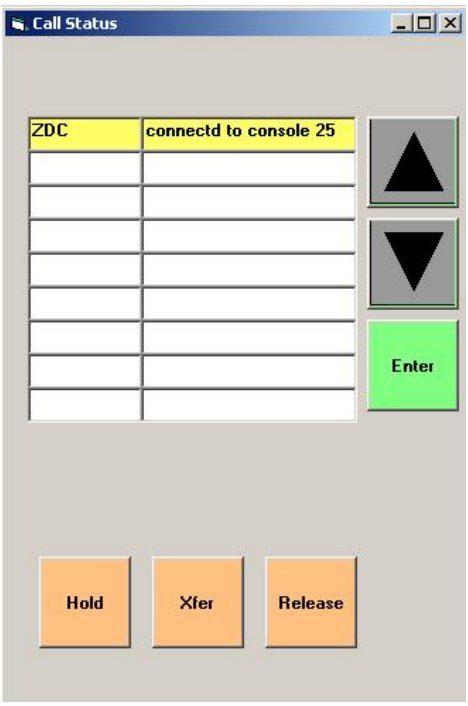


Figure 8. Call Status list.

4.2.5 Utility Functions

The nine horizontally arranged buttons at the top of the page in the TED-based interface (see Figure 4) allow the users to set their preferences and access the phonebook and call history features as described in Table 4. Except for the font size, screen controls, and volume buttons, these buttons have the same functions in the mouse-based interface (see Figure 5). In the mouse-based prototype, there are no buttons for adjusting the font, brightness, contrast, and volume. These are replaced by horizontal scroll controls placed in the right-hand section of the page that are always accessible.

Table 5. Utility Function Buttons for the TED-based Interface

Utility	Description
	The Screen Controls button allows users to adjust the brightness and contrast of the screen. Upon selection, the screen control opens two vertical scroll bars that allow users to independently adjust screen brightness and contrast. The blue buttons at the top indicate the current brightness and contrast setting. The screen brightness and contrast values range from 0-10.
	Upon selection, the Hset/Spk Vol button opens two vertical scroll bars so that users can individually adjust the headset and speaker volume. Also, the blue buttons at the top indicate the current value of the volume setting. The values range from 0-10.
	The Font Size button allows users to adjust the size of the text used for labels and captions on various buttons. This button when selected opens a vertical scroll bar using which users can adjust the font size. The size of the font is restricted by the size of the button. The selected font size is displayed on the top of this button.
	The Choose Home Pg Layout button allows users to browse through homepage layouts already being used and select a preferred layout for their own use. Selecting this button takes the user to a page where different homepage layouts are saved.
	The Return to Default Settings button resets the default settings for volume, screen brightness, contrast, font size, homepage layout, and so on.
	The Clean Display button primarily renders other buttons inactive while the TED is being cleaned. TEDs respond to touches from users, therefore, it is quite likely that users might inadvertently activate buttons while cleaning the display. The button toggles the display activation on and off. The button label toggles to “Clean Display On” when selected.

Table 5. Utility Function Buttons for the TED-based Interface (continued)

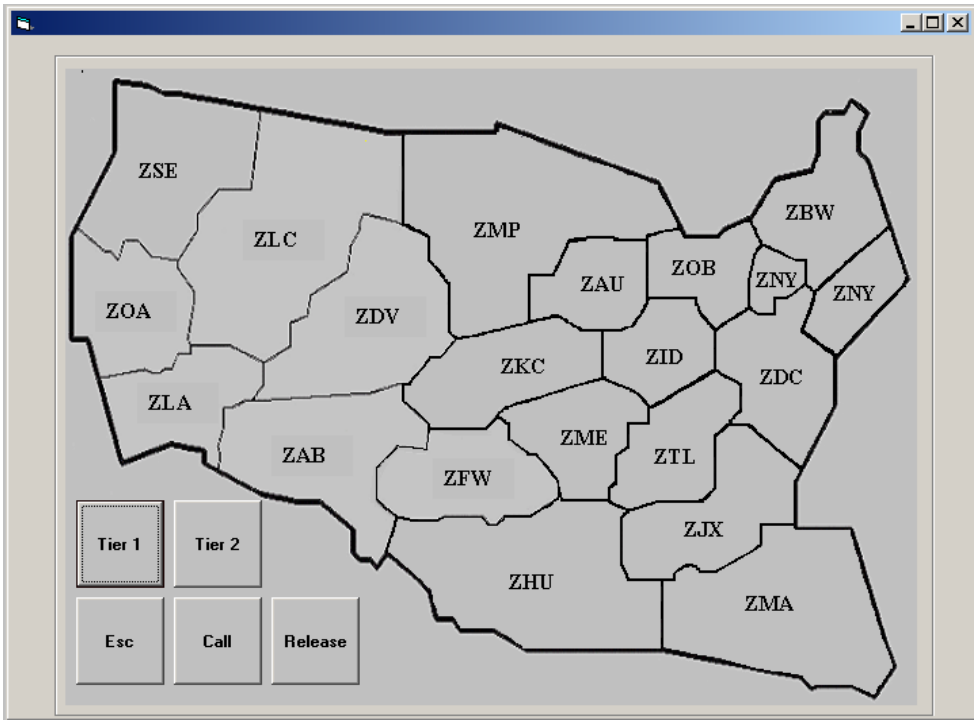
Utility	Description
Geo Layout	The Geo Layout button opens a geographic map of all the ARTCC boundaries (see Figure 9). Users can choose to call a center or a group of centers. To select a center and adjacent centers at the same time, the user first selects the intended region, for example ZDC, followed by selecting the Tier 1 or Tier 2 button to highlight adjacent centers. After selecting the center (s), the user then selects Call to initiate a two-party or a multi-party conference call. The Geo Layout button provides the ATCSCC controllers, at a glance, an easily interpretable graphic of which centers are connected in the conference using color codes. A green color indicates a center connected in the conference. There are some centers whose geographic size does not follow the minimum required (0.75 in. x 0.75 in.) standard for TED-based application explained in Section 4.1.2.  Figure 9. Map of air route traffic control center boundaries.

Table 5. Utility Function Buttons for the TED-based Interface (continued)

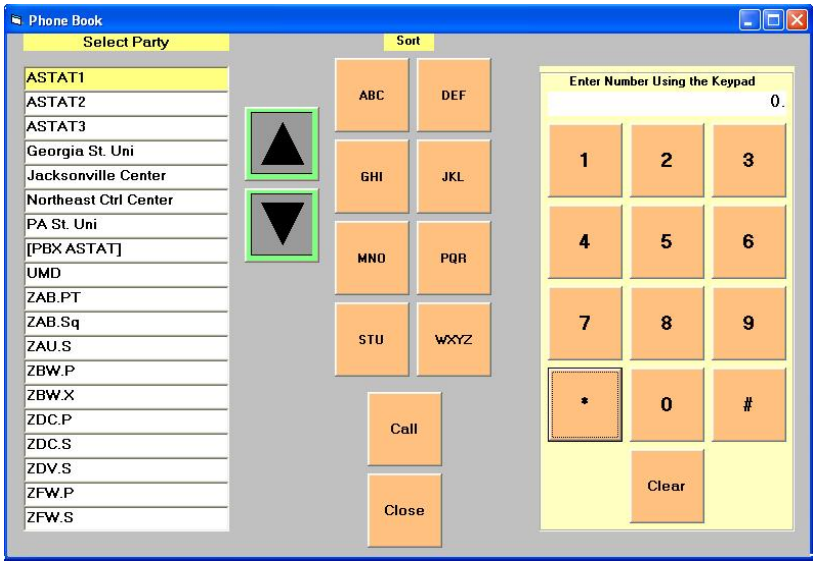
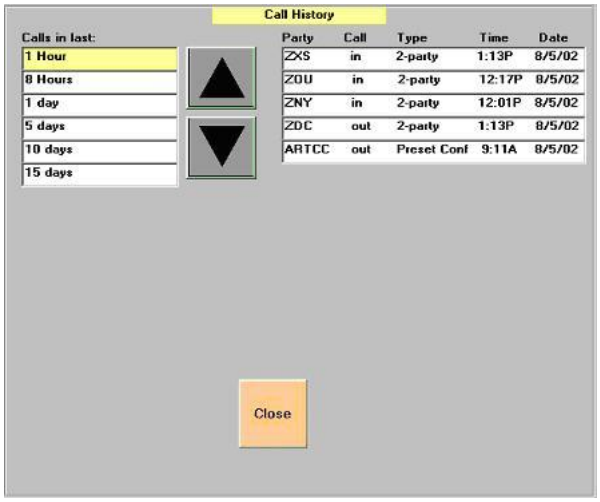
Utility	Description
Phone Book	Pressing or clicking the Phonebook button brings up the phonebook (see Figure 10), which lists all in-house and outside parties. It also provides a dial pad for dialing numbers that are not listed in the phonebook. To call a party, the user first highlights the party by using the up or down arrows and then pressing the Call button. Lists of parties, as in the case of the phonebook or call history pages, are necessary when the number of parties exceed the number of buttons that can be accommodated on a given page. Parties in a preset conference, for example, are also in lists. Lists can accommodate more parties in a limited amount of space. In the TED version choosing a party from the list could be tedious because the list items are too small to select. In the mouse-based version, a party can be chosen from the list by clicking on it. To address this issue, the TED version provides scroll buttons to scroll and highlight the parties whenever the parties appear in lists. We also provide alphabetized sort buttons, where necessary, so that users can directly scroll to the parties starting with a chosen alphabet set (see Figure 10).  Figure 10. Phonebook page.

Table 5. Utility Function Buttons for the TED-based Interface (continued)

Utility	Description
Call History	The Call History button allows users to view past incoming and outgoing call activity. Users can choose the history range to view the name, time, and type of inbound/outbound calls (see Figure 11). Controllers may refer to the log to check if a particular party participated in a conference, which is difficult to remember when a conference may include hundreds of participants.  Figure 11. Call history page showing time periods on the left and call details on right.

4.3 Conferences

Creating and managing conferences is the primary function of the CCS. In the sections that follow, we discuss how the prototype allows controllers to initiate and manage the major conference types.

4.3.1 Conference Phonebook Pages

Controllers can access a list of available conferences by pressing the Conf tab on the homepage. Like other phonebook pages, conferences can be arranged on hierarchical pages according to location, function, or other category defined by the ATCSCC. In the prototype, we organized the conferences on the main Conferences page by conference type (see Figure 12). Within the preset type, we organized conferences onto second level pages according to geographic location (e.g., ATCCC East) or operational function (e.g., contingency).

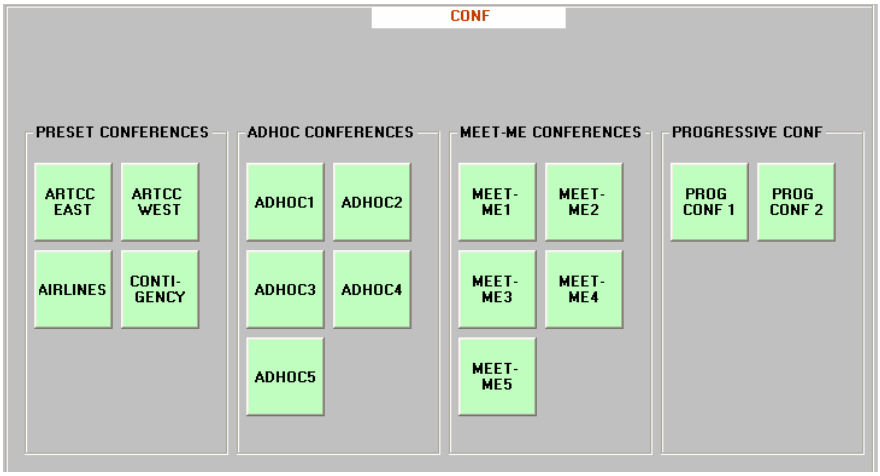


Figure 12. Conference Phonebook page showing buttons corresponding to individual conferences grouped into four overall conference types.

4.3.2 Preset Conference

A preset conference has a stable list of predetermined participants that is created by the ATCSCC and saved in the system. To access a preset conference, a user chooses it from the Conferences phonebook page. Depending on the number of preset conferences in the system, the user may need to navigate through several phonebook pages to find it. If a preset conference is commonly used at a console, a button for the conference may be configured onto that console's homepage. For example, in Figure 4, the button labeled "ZDC 1" corresponds to the ZDC First Tier conference, which Console 25 commonly uses. Other consoles can still access the ZDC 1 conference but would do so by navigating the phonebook pages.

Selecting a Preset Conference button displays the Conference Management page with the list of predetermined parties for the conference (see Figure 13). Table 6 describes the functions of each of the buttons.

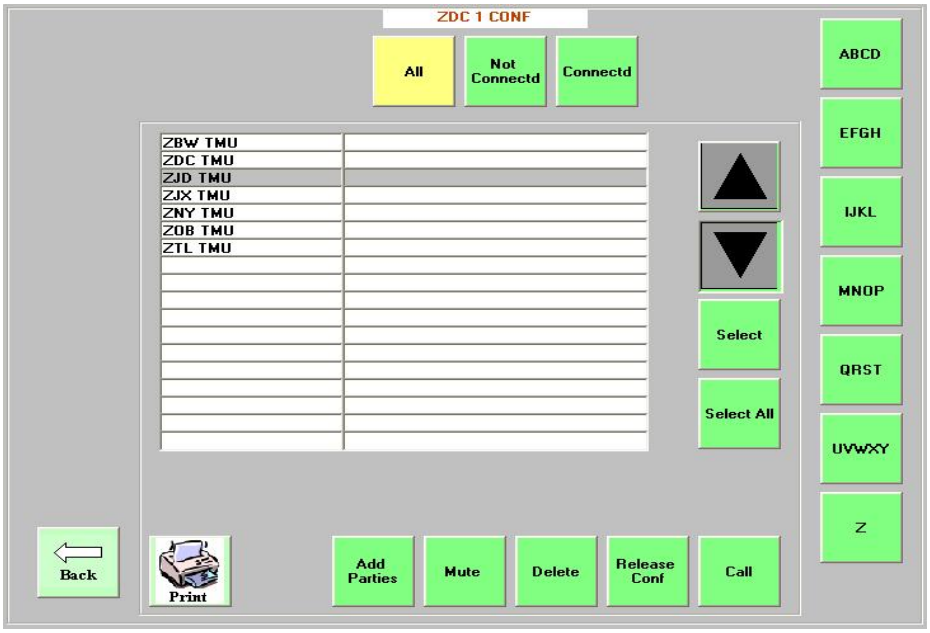


Figure 13. Conference management page for a preset conference.

Table 6. Functions of Buttons in the Preset Conference Page

Button	Function
All	Displays all parties associated with the conference regardless of whether they are connected to the conference.
Not Connectd	Displays only the parties that are not connected. Users may sometimes want to note the parties that did not participate in the conference. Before a conference begins, all parties are listed as not connected. As parties answer, they disappear from the not connected list and appear on the connected list.
Connectd	Displays only those parties that are connected so that the list reduces in size. As parties answer, they appear in the connected list. As parties hang up or are dropped from the conference, they disappear from the connected list.
Call	Initiates the preset conference. All associated parties will be dialed.
Release Conf	Releases all the participants from the conference.
Delete	Deletes a party from the current conference. This function does not delete the party from the saved participant list.

Table 6. Functions of Buttons in the Preset Conference Page (continued)

Mute	Mutes a connected party. To mute an active party, the user first highlights the party by using the scroll buttons in the TED-based version or by clicking on the party in the mouse-based version and then selecting the Mute button. A flickering asterisk (*) symbol next to the party in the list indicates that the party is currently speaking.
Add Parties	Used to add other parties to the current conference. This function does not add the party to the saved participant list. To add a party, a user first selects the Add Parties button followed by choosing a desired party by accessing the appropriate directory (e.g., to add Cincinnati tower to the list, the user first selects Add Parties followed by accessing the homepage and then selecting the CVG button.
Print	Used to print names of the parties in the list for future reference.

4.3.3 Ad hoc Conference

Unlike preset conferences, ad hoc conferences do not have predetermined lists of participants, and there is no concept of a list of parties who are not connected to the conference. Therefore, the conference management page is not populated with parties (see Figure 14) and does not contain the All, Connected, and Not Connected buttons.

To establish an ad hoc conference, controllers first select the Conf tab and then select an ad hoc conference button. Alternately, an ad hoc button may be available on the homepage if so configured.

Selecting an ad hoc conference prompts the user to assign a PIN for the conference. The Conference Management page appears once the PIN is assigned. The controller then selects the Activate Bridge button to allow the conference to begin receiving incoming calls. Upon selection, the button label changes to “Release Bridge.” Each participant dials in and enters the assigned PIN. The conference list populates as participants sign in. The functions (e.g., mute, delete, and release conference) available for the ad hoc conference are similar to those described for the preset conference. To deactivate the conference, the user selects the Release Bridge button.

In addition, the Answer button allows a controller to answer an incoming call and automatically add it to an ongoing ad hoc conference. This button saves the controller the alternative steps of leaving the conference page, answering the call, and then manually adding the participant to the conference.

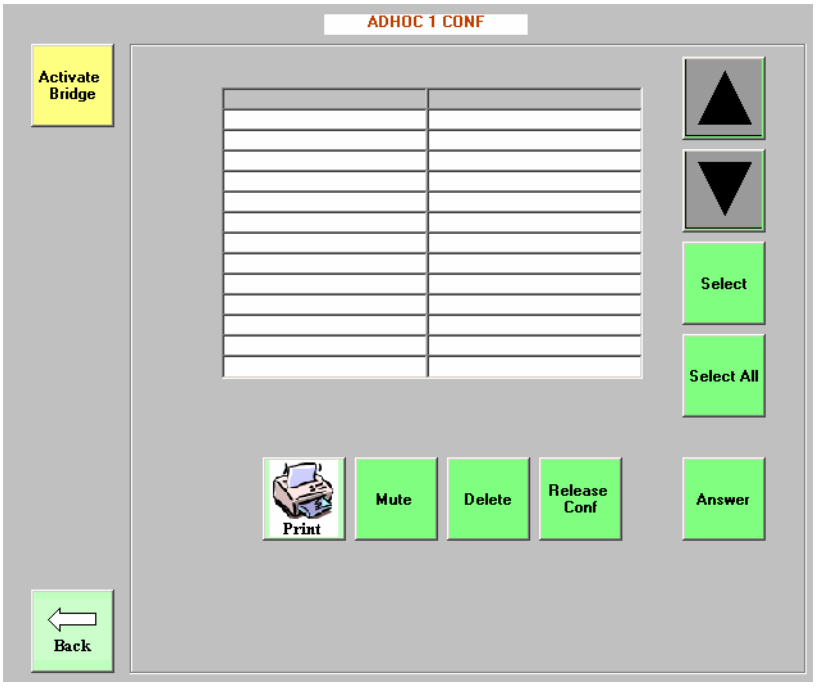


Figure 14. Conference management page for an ad hoc conference.

4.3.4 Meet-Me Conference

A meet-me conference includes a fixed set of participants who dial in using an assigned PIN. The PIN is assigned as part of configuring the conference so the controller does not need to establish a PIN for each instance of the conference. To activate a meet-me conference, the user selects the Conf tab and then selects the desired meet-me conference. Alternately, a Meet-Me button can be configured onto the homepage. Upon selecting the conference, a conference management page appears with the participant list populated, like a preset conference. Like an ad hoc conference, the controller must activate the bridge to allow conference participants to dial in to the conference. This is accomplished with the Activate Bridge button.

4.3.5 Progressive Conference

A progressive conference is initiated as soon as the user is connected to more than one party simultaneously. Managing a progressive conference is easiest when all parties appear on one page, such as the homepage. However, if some parties are not located on the homepage, substantial navigation is required for the controller to manage the conference. In this situation, the controller can select the Conf tab and select an ongoing progressive conference. The conference management page then appears, which allows the controller to see all participants at once and allows muting, deleting, and other important conference functions.

4.3.6 Conference Status Page

The Conference Status page is similar to the Call Status page in that it allows controllers to view all the ongoing conferences being hosted by the system (see Figure 15). The page can be accessed from the Current Conf button located at the bottom of the phonebook tabs if there is no conference ongoing at the console.

The page depicts each ongoing conference as a button. Buttons depict the name, host, and status of the conferences. The buttons can be used to transfer, mute, or release the whole conference on hold by selecting the appropriate telephone function button and then selecting the desired conference button.

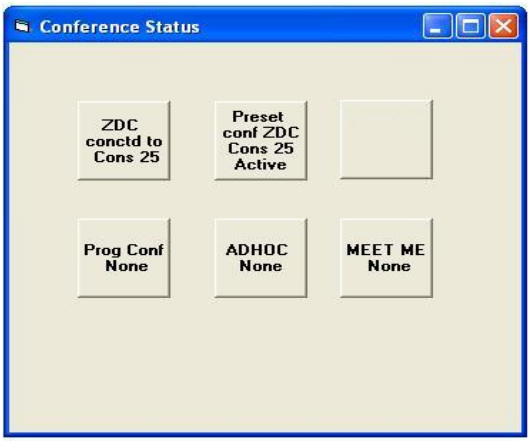


Figure 15. Conference Status page showing the type and status of conference.

5. CONCLUSION

This project employed an iterative prototyping approach to develop a CHI for the CCS.

Prototyping was useful because it allowed designers and users to visualize how a function might appear and examine how well it would support the users' tasks. We believe that the approach we adopted in this project had two advantages:

1. Prototyping allowed the human factors staff to better educate ourselves about the ATCSCC communication tasks, the users' priorities, and the operational environment. Better understanding of the task and priorities allowed us to provide better CHI designs.
2. The process educated the users of the system about human factors and usability issues. The users were in a better position to express their requirements to the CCS Program Office and identify usability issues with offeror's products once the vendor selection process began. They were also better prepared to serve as test participants in the formal

CCS acceptance and software testing because they understood human factors issues better.

Although the operational CCS has a somewhat different look and feel than the prototype discussed here, a number of concepts developed during prototyping were implemented in the CCS. None of these concepts is completely novel, but the prototyping process allowed the users to identify which functions and CHI elements were crucial to a successful CCS implementation. This allowed them to pursue these functions more aggressively in the formal requirements writing, vendor selection, and testing processes. Table 7 lists these concepts along and the underlying rationale.

Table 7. Prototype Concepts Used in the Operational CCS

Concept	Rationale
Homepage	The homepage is critical because it consolidates the parties that a position accesses regularly onto a single page. Without a homepage concept, the controller would need to navigate many pages in the phonebook to find the parties they need.
Buttons Indicate State	Each button that represents a party uses its label and its color to indicate what is currently happening with that party. This allows controllers to immediately know if the party is connected, on hold, ringing, and so on. Moving from the homepage to another page to determine the state of a party is cumbersome and can lead to mistakes. If the buttons do not indicate state, a controller is more likely to mistakenly dial a party that is already connected or forget that a party is on hold.
Geo Layout	The geographical layout displays a map of all the ARTCC boundaries (see Figure 9). The geo layout provides the controllers, at a glance, an easily interpretable presentation of which facilities are connected to a conference.
Room layout	The room layout feature displays the console layout in the room. This is useful when controllers want to transfer a call to a position, but they are not sure about the console number.

References

Ahlstrom, V. & Longo, K. (2003). *Human factors design standard* (Report Number HF-STD-001). Atlantic City International Airport, NJ: Federal Aviation Administration, William J. Hughes Technical Center.

Acronyms

ARTCC	Air Route Traffic Control Center
ATCS	Air Traffic Control Specialist
ATCSCC	Air Traffic Control System Command Center
BWI	Baltimore Washington International
CCS	Conference Control System
CHI	Computer-Human Interface
FAA	Federal Aviation Administration
HFDS	Human Factors Design Standard
NAS	National Airspace System
OTS	Operational Telephone System
PIN	Personal Identification Number
TED	Touch Entry Display
TMU	Traffic Management Units
TRACON	Terminal Radar Approach Control
ZDC	Washington ARTCC

Appendix A

CCS Prototype Demonstration Script

DIRECTIONS: complete each step as presented in the script. If you are unable to perform a step or have any concerns, please approach one of the Human Factors Engineers.

Note: Icons at top right corner of the homepage are not part of the CCS interface and only facilitate in initiating an incoming call, demonstrating auto transfer feature and terminating the application.

1. ❑ Call anyone of these parties: BWI, CVG, and DAY from the homepage.
2. ❑ Release the above call.
3. ❑ Answer the first incoming call (ZDC).
4. ❑ After answering, place the call on hold.
5. ❑ Answer the second incoming call.
6. ❑ Transfer the current call to console 15 using your preferred way of transferring a call.
7. ❑ Retrieve the first call from the hold list and initiate a progressive conference by calling UPS and Fedex Airlines.
8. ❑ Release the progressive conference.
9. ❑ Initiate ZDC first tier preset conference and perform the following operations.

    - Add & Delete Parties--- **Add ZAU, Delete ZBW ---- Ground Stop IAD**
    - Call
    - Mute (party with flickering "*")
    - Hold ZOB TMU
    - Release conference

10. ❑ Initiate AD HOC—**Runway closure event.**

    1. Perform the following operations

        - Assign pin "8268"
        - Activate Bridge
        - Answer call from "ZNY"
        - Delete "ZNY"

    2. Release Bridge/Terminate conference

11. ❑ Initiate Meet Me—**Strategic Team Planning telcon.**

    - Perform the following operations
    - Initiate Conference
    - Release conference
    - Answer incoming call from "ZNY" to bring it in the Meet Me conference
    - Release conference

12. ❑ Barge into an existing conference at Position 14.

3. (To simulate this, first release all conferences. Then initiate a ZDC first tier preset conference. Now perform the barge operation)

13. ❑ Override position 15.

14. ❑ Forward position—**Controller leaving the position- fwd to position 14.**

15. ❑ Initiate auto-transfer request for the incoming call party ZNY.

4. (Make sure you have an incoming call flashing. If not click the telephone icon in the top right corner of the homepage to initiate incoming calls)

16. ❑ Add/Remove temp parties to homepage---Add N90, then remove temporary party N90.

17. ❑ Check calls made/received in the past 8 hours.

18. ❑ Browse through the different homepage layouts and ring tones available by accessing the appropriate screen.

19. ❑ Change screen brightness to 8 and contrast to 10.

20. ❑ Change font size to 10.

Appendix B

CCS Prototype Installation Directions

The CCS prototype is a Microsoft Windows®-based program containing multiple pages with data displays and buttons. The following steps describe installing and operating the prototype.

Installation:

1. Open the contents of the CD by choosing the appropriate drive.
2. Double click on the Demo.exe file to open the TED-based version of the prototype or mouse-based.exe to open the mouse-based version.
3. The program will run automatically.

Operation:

1. By default, the homepage is first displayed upon opening the executable (exe) file.
2. Interact with different functions available in the prototype by selecting buttons via the TED or clicking with the mouse. The script (Appendix A) will guide users through the different functions.
3. To exit the prototype, press the Stop icon button located at the top right corner of the screen.